TELKOMNIKA, Vol.14, No.1, March 2016, pp. 219~227
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v14i1.1926

Stemming Influence on Similarity Detection of Abstract Written in Indonesia

Tari Mardiana*[1], Teguh Bharata Adji[2], Indriana Hidayah[3]
[1,2,3]Departement of Electrical Engineering and Information Technology, UGM,
Jalan Grafika No. 2, Yogyakarta, Indonesia
[1]Informatics Engineering, Tanjungpura University,
Jalan Jenderal Ahmad Yani, Pontianak, Indonesia
*Corresponding author, email: tari.mardiana@gmail.com[1], adji.tba@gmail.com[2], indriana.h@ugm.ac.id[3]

Abstract

In this paper we would like to discuss about stemming effect by using Nazief and Adriani algorithm against similarity detection result of Indonesian written abstract. The contents of the publication abstract similarity detection can be used as an early indication of whether or not the act of plagiarism in a writing. Mostly in processing the text adding a pre-process, one of it which is called a stemming by changing the word into the root word in order to maximize the searching process. The result of stemming process will be changed as a certain word n-gram set then applied an analysis of similarity using Fingerprint Matching to perform similarity matching between text. Based on the F_1-score which used to balance the precision and recall number, the detection that implements stemming and stopword removal has a better result in detecting similarity between the text with an average is 42%. It is higher comparing to the similarity detection by using only stemming process (31%) or the one that was done without involving the text pre-process (34%) while applying the bigram.

Keywords: *abstract, Indonesian, similarity, stemming, word*

1. Introduction

Indonesian can be regarded as a high-context language because of the complexity use of words in it. In the field of Natural Language Processing (NLP), Indonesian is included in less-resourced language so that the research as regards this language is limited. Indonesian words rarely have the precise form because of the use of affixes among stem words, either prefixes, suffixes or repetitive word can change the meaning of the word itself. There is a need to conduct a process to change the words contain affixes becomes the root form of the word so that the word meaning is easy to understand through the process called stemming. Stemming is almost similar to lemmatization but stemming does not need to pay attention to the meaning of the word formed from the process.

The stemming algorithm is different for one language to another so that the implementation of the same technique to other language can lead to the different result. It is because the stemming is language-dependent process. As the phase of pre-processing in text retrieval, one of popular stemmer, Porter algorithm is implemented not only to processing text in English but also Spain and Portuguese [1, 2]. To make the Porter algorithm being able to be applied in Indonesian, modification has been done by adding some rules and measurement requirenment by [3]. There are some other kinds of stemming algorithms that has been proposed by [4] to be implemented in Indonesian, such as Nazief and Adriani Algorithm, Arifin and Setiono Algorithm, Vega Algorithm, Ahmad, Yusof, and Sembok Algorithm, and Idris Algorithm. A research [5] that compared between Porter and Nazief Adriani (NA) algorithm which implemented in Indonesian document shows that Porter algorithm has the less precision value in result. Another research related to this fact says that NA algorithm is able to conduct stemming process with up to 93% of success [6]. However, there is no provision which algorithm should be applied in order stemming process can provide better result to improve the accuracy of Information Retrieval since basically, a document is seen as text that coherence and contain useful information.

Received April 25, 2015; Revised December 3, 2015; Accepted December 26, 2015

Duplication in a text document can be in terms of sentences that has been arranged and modified into smaller part of words. To find out a couple of similar words, it is necessary to divide the sentence into smaller fragment. The word forming process that used to find the key-word of a sentence can be through text-preprocessing that consist of keyphrase matching [7] and tokenization [8]. The state of art in doing the similarity matching is the duplication that exist in the text must be detecting and analyzing although it is only a small part of word that has been modified which is text re-use. The implementation of stemming process in order to detect similarity by using statistic is believed as one way to compare the words among different texts and to identify the identical words which have the same meaning [9]. Indentifying similarity is usually done based on the word similarity, fingerprint similarity or latent semantic analysis (LSA) [10].

There is no previous study related to the stemming process in Indonesian as a part of the text pre-processing, especially the one about similarity detection in written abstract. The detecting process involving the stemming can lead to erasing important information, so it is necessary to do a further study to see how important the text pre-processing process in similarity detection. This paper aims to see the influence of stemming process in detecting similarity by comparing the matching result between the abstract that implemented the stemming process and the ones are not. This study uses the Nazief and Adriani algorithm and n-gram to process the order word stem from the stemming result, the value of similarity was taken based on the fingerprint match that is done to see whether there is a similarity exists among the abstracts or not. Then, according to that analysis, this paper explains the probable factors that affect the detection result and the fallacy factor that probably cause the failure in stemming process.

2. Research Method

In detecting text similarity written abstract, there were some phases that should be done in this research, including preprocessing input abstract, tokenization, hashing text, and similarity analysis, with output of similarity value produced between text. This following figure (Figure 1) shows the flow of the phases.

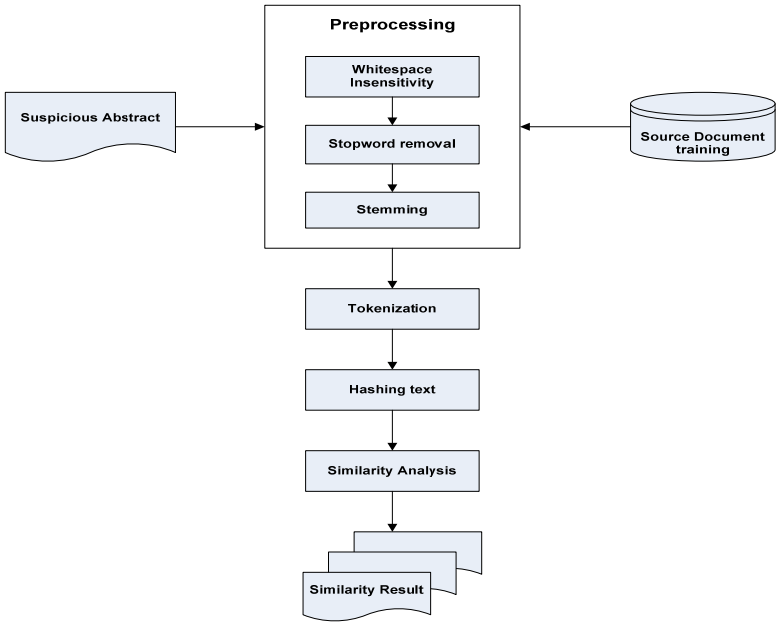

Figure 1. System flows of similarity detection

Abstract that has been passed prepocessing will be formed as word gram term set according to gram value that has been determined before through tokenization process. By

using hashing function, each gram set will be represented as hexadecimal and used as unique identity for matching up. Detail explaination of every process according below.

2.1. Preprocessing

The very first step before document or text is processed furthermore, consists of whitespace insensitivity process, stopword removal, and stemming.

a) *Whitespace insensitivity*; This process will eliminate all unnecessary punctuation such as space, colon, semicolon, numbers, etc and case folding to change all text characters become lowercase.
b) *Stopword Removal;* This process will eliminate common word that is often used or meaningless repetitive word in sentences, for example “masing-masing”. This process will result unique word set which is expected to improve the accuracy of similarity [8].
c) *Stemming;* This process will change the corespondent words in the same root word, for example “membeli” and “dibeli” come from the root “beli”. In indonesian there are so many words which contain affix, either prefix, infix, sufix, confix, or word repetiton inserted in a stem. Through stemming process, the text will only contain root words in order to perform plagiarism detection regards to the text which has been changed in words order.

2.1.1. Stemmer Nazief Adriani

In Indonesian morphology, there are some rules of word formation that contains inflection and derivative word as defined in the following rules.

Inflictional = (root+possesive_pronouns) | *(root+particle)* | *(root+possesive_pronouns+particle)*
Derivational = prefixed | *suffixed* | *confixed* | *double_ prefixed*

in which:

Prefixed = *prefix + root;*
Suffixed = *root + suffix;*
Confixed = *prefix + root + suffix;*
Double_prefixed = *(prefix + prefixed)* | *(prefix + confixed)* | *(prefix+prefixed+suffix)*

From the definition above, in general Indonesian morphological structure as described in [3] can be defined as the following rules below.

[prefix1] + [prefix2] + root + [suffix] + [possesive_pronouns] + [particle]

In addition, some prefixes such as *ber-, meng-, peng-, per-, ter-* will change from their origins, called Nasal Subtitution. It namely the state of articulation that changed when one prefix (e.g. *meng-*) inserted in the stem of the word [11] as can be seen in Table 1. These changes greatly depending on the inserted after the first word affixes.

Table 1. Examples of word formation rules with prefix *–meng*

Prefix	Origin	Formation	Subtitution
meng-	tulis	menulis	{meng|t} = *men-*
meng-	sewa	menyewa	{meng|s} = *meny-*
meng-	pakai	memakai	{meng|p} = *mem-*
meng-	kritik	mengkritik	{meng|k} = *meng-*

Indonesian morphology is more difficult and complicated than English since during the morphological process, Indonesian always combine among affixes, root word and grammatical rules at the same time.

Nazief and Adriani (NA) algorithm was first introduced in 1996 in a technical report from University of Indonesia and was further developed in the study [12]. This algorithm is based on morphological rules are interlinked and grouped together, and then encapsulated as allowed part of the word and not include affixes such as prefixes, suffixes, and confixes to get the root of

a word. Basically, all stemming are able to increase the sensitivity of the retrieval of documents, however by doing a search root word through stemming often lead to the removal of the meaning of the word itself. If the removal rule is done according to the determined order it would be sure capable of preventing any overstemming that is a condition of over word removal or understemming that is the word cannot be performed for stemming because it can't see the removal rule of the word, so that the failure that lead to the stem cannot be found able to minimize. Basic removal process of NA stemmer based on the explanation in study [6] can be seen in Figure 2.

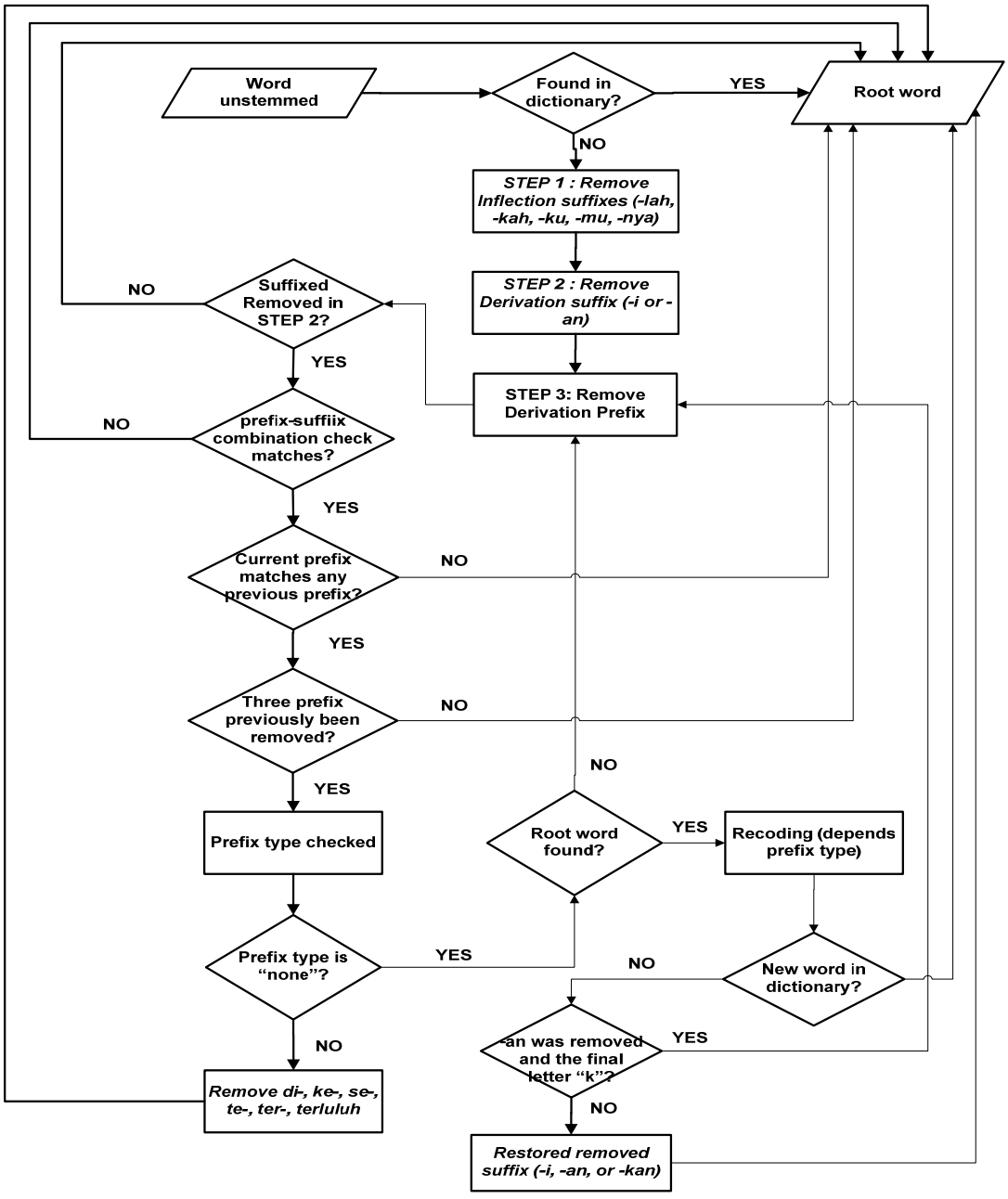

Figure 2. Basic removal process of NA stemmer

Nazief and Adriani algorithm is applying 37 rules that can be used to perform word stemming. The performance of this algorithm is based on three parts, they are grouping affixes,

usage rules and the establishment of limits, and the dictionary is used. Dictionary becomes an important part because it is used to check whether a word has met its stem or not. Before the affix removal process, there are several things that must be considered in the uses of this algorithm.

a) **Inflection suffixes**: a set of suffixes that doesn't change the stem, such as *-lah, -kah, -ku, -mu, -pun, -nya*. Particles including *-lah, -kah, -tah* or *–pun*, and possesive pronouns including *-ku, -mu, -nya*.
b) **Derivation suffixes**: a set of suffixes are directly placed in the root, but have more than one suffixes, such *as -i, -an, -kan*.
c) **Derivation prefixes**: prefix that is attached directly to the stem or words that have two prefixes are placed together, such as *di-, ke-, se-, te-, be-, me-,* and *pe-*.
d) **Prefix Disallowed Sufixes**: a combination of the prefix and suffix that are not allowed to be attached to the stem as in Table 2.

2.2. Tokenization

Is an initialization phase by conducting a structured text extraction in the form of a single word. In this stage, the abstract text will be established as a set of shingles using the word n-gram (WNG). WNG is one style of construction models which can be used as a way of verifying the detection process [13]. The longer the value of n is used the less set shingles that are formed. N-gram approach in research is urgently need proper value of n in order to produce a clear distinction between the sentences in the document [14]. The examples of tokenization with WNG showed in Table 3.

Table 2. The combination of the prefix and suffix disallowed

Prefix	Suffix
be-	*-i*
di-	*-an*
ke-	*-i, -kan*
me-	*-an*
se-	*-i, -kan*

Table 3. Examples formation of shingles with WNG

Original	belajar komputer itu tidak sulit
Unigram	{belajar}{komputer}{itu}{tidak}{sulit}
Bigram	{belajarkomputer}{komputeritu}{itutidak}{tidaksulit}
Trigram	{belajarkomputeritu}{komputeritutidak}{itutidaksulit}
Fourgram	{belajarkomputeritutidak}{komputeritutidaksulit}

2.3. Hashing Text

In this process, all shingles set will be represented as groups of hexadecimal called fingerprint through hash function. The objection to perform hashing is to obtain unique values as identity to differ each formed words. Fingerprint is one of the techniques that can be used to perform similarity analysis that can be lead to plagiarism act [15].

2.4. Similarity Analysis

The last process that done through matching the formulated fingerprint value through hashing process between the abstract which is indicated as a plagiat contrast to the abstracts that have been in the database. The value of the similarity among the number of A and B shingles which has a resemblance union C will be calculated using the Dice coefficient according formula (1) and expressed as a percentage.

$$similarity = \frac{2C}{A+B} = \frac{2(A \cap B)}{|A|+|B|} \quad (1)$$

To support the process and testing, it is used a dataset consisting of 30 data of abstract documents in the field of Information Technology. As many as 25 data of training materials and testing as many as 5 data. The number of stem in a dictionary word that is used to assist the process of stemming as much as 31.296 words, while the number of stoplist are available to perform stopword removal as many as 756 words. As the further evaluation, the fault cases that commonly happened while applying the NA algorithm are classified by Jelita et al [6] into some categories: 1) Non-root words in dictionary, 2) Hypenated words, 3) Incomplete Dictionary, 4) Misspellings, 5) Incomplete affix rules, 6) Overstemming, 7) People's names, 8) Combined Words, 9) Recoding ambiguity (dictionary related), 10) Acronyms, 11) Recoding ambiguity (rule related), 12) Other, 13) Understemming, 14) Foreign words, and 15) uman Error.

The counting on the stemming accuracy value was done based on the number of suceed word (S_B) divided by the total of unique words in text (S_T) according to formula (2). The evaluation of succesful detection will be based on precision value and recall which are got from (3) and (4) equotations. F_1-score is used to balance the number of precision and the recall which counted by using the (5) equation.

$$Accuracy = (S_B / S_T) x100\% \quad (2)$$

$$precision = \frac{TP}{TP + FP} \quad (3)$$

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \quad (4)$$

$$F_1 - score = \frac{2.p.r}{p + r} \quad (5)$$

3. Results and Discussion

This section shows the result of the testing and the analysing toward stemmer algorithm usage and similarity detection system that have been constructed. Scenario testing is divided into two parts: the first test to see the success of the word stemming in text by using NA algorithm and the second is detection evaluation which is done by analyzing the similarity based on the number of similarity resulted from the text implemented the stemming process and the ones which did not.

The first test was carried out to test the performance of the algorithm stemmer (NA) to perform stemming to 10 abstract. Prior to stemming, stopword removal will be done to get rid of common words in Indonesian are considered to have no significance, such as "di", "yang", "adalah", etc so the only remaining unique words alone. Table 4 shows the results of word root stemming using the NA algorithms and Table 5 shows the classification on the fault cases (stemming errors) that occured in the abstract document.

Table 4. The result toward the word stemming using NA algorithm

Abstract Doc	Word Count	Unique Word	Correct Stem	
			NA	Accuracy (%)
1_ABS.docx	248	161	158	98.14
2_ABS.docx	263	190	183	96.32
7_ABS.docx	330	232	222	95.69
9_ABS.docx	164	117	114	97.43
10_ABS.docx	207	143	141	98.60
12_ABS.docx	287	217	212	97.69
13_ABS.docx	244	175	171	97.71
14_ABS.docx	286	198	190	95.95
15_ABS.docx	281	218	195	89.45
20_ABS.docx	126	85	83	97.65
	AVERAGE			96.46

Table 5. Most Fault Cases in Abstract

Fault cases	Total Case(s)	Fault cases	Total Case(s)
Non-root words in dictionary	4	*Recoding ambiguity (dictionary related)*	13
Hypenated words	8	*Recoding ambiguity (rule related)*	8
Incomplete Dictionary	4	*Other*	2
Misspellings	2	*Understemming*	1
Incomplete affix rules	10	*Foreign words*	2
Overstemming	6	*Human Error*	5
Combined Words	2	**TOTAL**	67

Several factors must be noted in stemming failure of abstract document such as failure classification based on Table 5 with detail below.

a) Some words consider as foreign word, for example *meminimalisir, normalisasi* encounter the stemming failure because they do not include the word removal rule. While the words *sedangkan, pencari, pelaku, diolah* encounter overstemming into *dang, pencar, pela*, and *o*. Overstemming can occur because the process of removing affixes as much as possible according to the rule applied. Another stemming failure is usage of uncommon words in abstract, eg. *kerapkali.*
b) The most prevalent cases of word stemming failure are the words include in recording ambiguity (dictionary related). In NA algorithm that is accordance to dictionary as a base for stem matching, for example removal rule of words that contain confixes *per-* and *-an* when encountering a stem begins with an *(r)* lead to stemming failure, eg. *perawatan* becomes *awat* and *perancangan* becomes *ancang*. It because in the words dictionary there are words *rancang* and *ancang*, *rawat* and *awat*, and some other words.
c) The words refer to quantity, for example *sejumlah* and *berjumlah* are subjected to inappropriate inflectional suffixes removal rule for *-lah* that lead to failure in the process. Besides, there are still many errors in stemming for repetitive words, eg. *sehari-hari, berbeda-beda.*
d) The failure due to human error is encountered in cases where the words are typed with no space so that two words written as if they are one word.

In the second test aims to detected the similarity that emphasized on the preprocessing text in terms of pure stemming (ST), combining stemming and stopword removal (ST+SWR), and detected without the preprocessing which is done by checking the success of the measurement using precision and recall value. The length values of the word n-gram (WNG) that used in this study in order to construct word term are 2, 3 and 4. Five abstracts were randomly chosen and made as the testing samples to evaluate the work of the detection system that is constructed. The accuracy number of the precision and the recall value along with the some script can be seen in Figure 3 and 4.

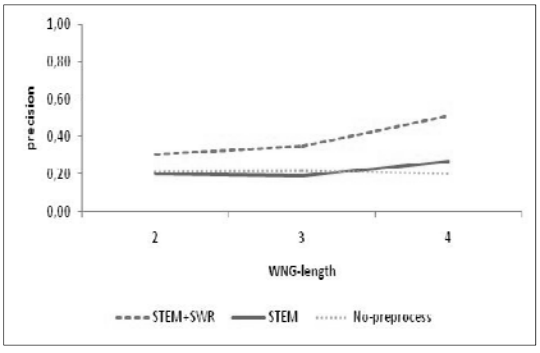

Figure 3. Evaluation of similarity detection based on precision

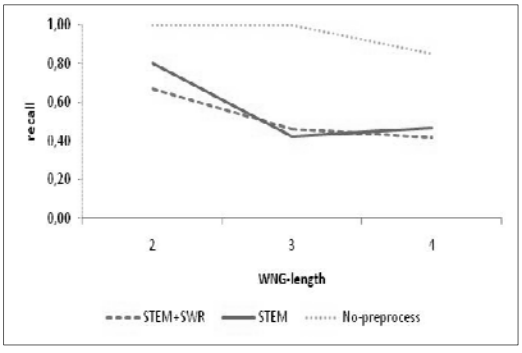

Figure 4. Evaluation of similarity detection based on recall

Based on Figure 3 and 4, we can see that the similarity detection in the abstract that being the subject of the stemming process and the stopword removal during the preprocessing text have higher precision value compare to the ones that only used the stemming process without the preprocess with a percentage of 67% while using the fourgram. Though, the best recall value is showed in the detection process that excluded the preprocessing with the number of percentage 100% while using the bigram and trigram. This result supports the research result done by [4] in which it said that the effect of stemming in the text retrieval is considered as a help to increase the recall value but it reduces the precision value. There are some other false-positive cases that being the causes of the precision low value.

The similarity value resulted from the inexistence of the preprocessing with the low value of gram (bigram) gave a higher result compared to the ones using the preprocessing with the rate of 4.75%. The low similarity value shows that each of the final project abstracts is categorized as a unique text since they have a different content composition that differentiated based on their research field. This condition shows that similarity is not a significant way to determine reduplication, though it is still can be used as the first filter before doing similarity detection on a written discourse. Based on Bazdaric assumption [16], plagiarism in a piece of writing is estimated to have the range of 5-10 % similarity or around 100 words similar in one document, so we also have to pay attention on the size of the document checked.

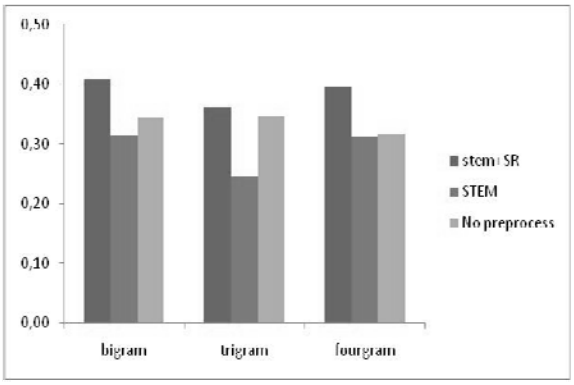

Figure 5. Evaluation of similarity detection based on F_1-score

From the overall result showed in Figure 5, text preprocessing on similarity detection that combines stemming and stopword gives the highest accuracy value to all word gram level that applied in 2 scenarios which are evaluated with the rate of F_1 number is 42%. The fault factors that occurred during the process of the word stemming make the text containing some meaningless and unsuitable words so it reduces the number of similarity that being a section of two texts, especially when the result is compared to the ones experienced detection. The addition of stopword removal is able to increase the work of stemming by reducing the number of the words that commonly repeated, if we compare it to the ones that only experienced the stemming process without preprocessing. So, it is necessary to be further discussed whenever it is applied in the detection that involves statistic words. This result also shows that combining two of them (stemming and stopword removal) can give a better result in the similarity detection for the more unique words existing in the text that are needed to be checked. There is a big possibility that even the smallest modification can be detected by using this process. The support toward the implementation of WNG which pay no attention on the word position/ term in the searching words also help the detection which involve the case of position changing. However, the application of document fingerprint as the matching tool which based on the length of word gram also need to be seen because it will make the reduplicated part of the text is clearly seen and on the other way around.

4. Conclusion and Future Work

Based on the previous study, Nazief and Adriani algorithm is considered as the qualified one to do the good stemming process although there are still some mistakes in the words. This problem leads to a condition where it is definitely important to create a dictionary that contains the root word. It is also necessary to make a dictionary that contain the standardize root word in KBBI (Kamus Besar Bahasa Indonesia) which can fulfill the necessity of the users. The implementation of text preprocess to detect similarity in Indonesian written abstracts is going to be more suitable whenever it is applied the stemming process along with the stopword removal process because their combination can make the unique words which resulted from the preprocess to help the matching process that involve the position or term changing. However, the similarity detection without the preprocess is still applicable as the alternative way since the measurement is done based on the intersecting words exist in the compared text. Besides, the value of similarity is not adequate to decide that an abstract is a result of reduplication or not. Though, it still can be used as a standard to know whether some parts of the abstract are reduplicated by the other abstract or not.

For the further research, it is suggested to focus on solving the fault cases that found during the stemming process, such as the one related to the non-root words in the dictionary and hyphenated words in order to reduce the fallacy in the stemming process. The similarity detection toward abstract of a research is highly advised to be focused more on the semantic part in order to cover the weakness of the words matching that commonly only done by finding the similarity in the word term while there is a possibility that words contain different meaning based on the field of the research.

References

[1] CG Figuerola, R Gómez, EL De San Román. Stemming and n-grams in Spanish: An evaluation of their impact on information retrieval. *Journal of Information Science*. 2000; 26(6): 461-467.
[2] MVB Soares, RC Prati, MC Monard. Improvement on the Porter's Stemming Algorithm for Portuguese. *Latin America Transactions, IEEE (Revista IEEE America Latina)*. 2009; 7(4): 472-477.
[3] FZ Tala. Effects on Information Retrieval in Bahasa Indonesia. Master Thesis. University of Amsterdam, Institute for Logic, Language and Computation; 2003.
[4] J Asian. Effective Techniques for Indonesian Text Retrieval. PhD Thesis. RMIT University, Melbourne, School of Computer Science and Information Technology, Science, Engineering, and Technology Portfolio; 2007.
[5] L Agusta. *Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia.* Presented at the Konferensi Nasional Sistem dan Informatika, Bali. 2009: 196-201.
[6] J Asian, HE Williams, SMM Tahaghoghi. *Stemming Indonesian*. in Twenty-Eighth Australasian Computer Science Conference (ACSC2005). Newcastle, Australia. 2005; 38: 307-314.
[7] I Veritawati, I Wasito, T Basaruddin. Text Preprocessing using Annotated Suffix Tree with Matching Keyphrase. *International Journal of Electrical and Computer Engineering (IJECE)*. 2015; 5(3): 409-420.
[8] Z Ceska, C Fox. *The influence of text pre-processing on plagiarism detection.* in International Conference Recent Advances in Natural Language Processing, RANLP. 2009: 55-59.
[9] S Hariharan. Automatic Plagiarism Detection Using Similarity Analysis. *The International Arab Journal of Information Technology*. 2012; 9(4): 322-326.
[10] Z Alfikri, A Purwarianti. Detailed Analysis of Extrinsic Plagiarism Detection System Using Machine Learning Approach Naive Bayes and SVM. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(11): 7884-7894.
[11] M Haspelmath, AD Sims. Understanding Morphology. 2nd ed. London: Hodder Education, an Hachette UK Company. 2010.
[12] M Adriani, J Asian, B Nazief, SM Tahaghoghi, HE Williams. Stemming Indonesian: A confix-stripping approach. *ACM Transactions on Asian Language Information Processing (TALIP)*. 2007; 6(4): 1-33.
[13] B Stein. Technology for Text Plagiarism Analysis. Bauhaus-Universität Weimar. 2010.
[14] AZ Broder. Syntactic clustering of the Web. *Computer Networks*. 1997; 29(8-13): 1157-1166.
[15] B Stein, SM Eissen. *Near similarity search and plagiarism analysis.* in From Data and Information Analysis to Knowledge Engineering. Springer. 2006: 430-437.
[16] K Baždarić. Plagiarism detection – quality management tool for all scientific journals. *Croatian Medical Journal*. 2012; 53(1): 1-3.